

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-314227

(43)公開日 平成6年(1994)11月8日

(51)Int.Cl.⁵

G 0 6 F 12/00

識別記号

庁内整理番号

F I

技術表示箇所

5 3 5 Z 8944-5B

5 1 8 A 8944-5B

審査請求 有 請求項の数14 OL (全 18 頁)

(21)出願番号 特願平6-39531

(22)出願日 平成6年(1994)3月10日

(31)優先権主張番号 0 5 6 0 1 3

(32)優先日 1993年4月30日

(33)優先権主張国 米国(US)

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 シャウーベン・シ

アメリカ合衆国78726、テキサス州オース
ティン、クロスランド・ドライブ 11033

(74)代理人 弁理士 合田 潔 (外2名)

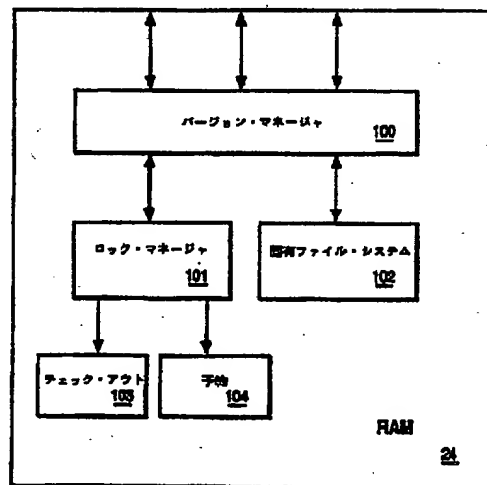
最終頁に続く

(54)【発明の名称】 バージョン化オブジェクトに対するロッキング機構

(57)【要約】

【目的】 分散データ処理環境において、複数のユーザ
によるバージョン化オブジェクトへの同時アクセスを制
御する方法、及びシステム。

【構成】 最初にユーザがアクセスを希望するデータ・
セットの第1の部分に対する書き込みロックを探索する。
書き込みロックが検出されない場合、第1の部分に対する
書き込みアクセスを許可することにより要求が許可され
る。書き込みロックが設定されている場合、他のユーザが
それを変更することを禁止するが、他のユーザは第1の
部分への読出しアクセスは許可される。第2のユーザが
データ・セットの第2の部分へのアクセスを希望し、第
2の部分が第1の部分と幾つかの共通要素と、共有され
ない要素を有するとき、第2のユーザの要求は、第1の
部分と共有されない第2の部分の1部だけに対し、部分
的に許可される。



【特許請求の範囲】

【請求項1】 コンピュータ・システムのメモリ内の複数のサブセットから成るデータ・セットへの複数のユーザによる同時アクセスを制御する方法であって、

前記データ・セットの第1のサブセットへの書き込みロックを探索するステップと、

第1のユーザによる前記第1のサブセットへの書き込みアクセスのための第1の要求を許可するステップと、

他のユーザが前記第1のサブセットを変更することを禁止するために、前記データ・セットの前記第1のサブセ

10

ットに書き込みロックを設定し、他のユーザによる前記第1のサブセットへの読出しアクセスは許可するステップと、

を含む方法。

【請求項2】 前記データ・セットの第2のサブセットへの書き込みロックを探索するステップと、

第2のユーザによる前記第2のサブセットへの書き込みアクセスのための第2の要求を許可するステップと、

他のユーザが前記第2のサブセットを変更することを禁止するために、前記データ・セットの前記第2のサブセ

20

ットに書き込みロックを設定し、他のユーザによる前記第2のサブセットへの読出しアクセスは許可するステップと、

を含む、請求項1記載の方法。

【請求項3】 前記第1のサブセット内の要素の第1のセット、及び第1のサブセット外の要素の第2のセットを有する、前記データ・セットの第3のサブセットへのロ

30

ックを探索するステップと、

第3のユーザによる前記第3のサブセットの要素の前記第2のセットへの書き込みアクセスのための第3の要求を許可するステップと、

前記第3のサブセットの要素の第2のセットへの書き込みロックを設定するステップと、

を含む、請求項1記載の方法。

【請求項4】 前記第1のユーザからのコマンドにตอบสนองして、前記第1のサブセットを変更するステップと、

前記第1のサブセット内の変更を反映するために、前記データ・セットを更新するステップと、

40

前記第1のサブセットへの書き込みロックを解除するステップと、

を含む、請求項1記載の方法。

【請求項5】 前記第1のサブセットへの書き込みロックが通知モードであり、

前記第2のユーザによる前記第1のサブセットへの書き込みアクセスのための前記第2の要求にตอบสนองして、前記第1のサブセットへの書き込みロックを探索するステップと、

前記第1のユーザに、前記第1のサブセットに対する前記第2の要求が受信されたことを示すメッセージを送信するステップと、

所定期間内に前記第1のユーザによる応答が受信されない場合、前記第1のサブセットへの書き込みロックを解除するステップと、

を含む、請求項1記載の方法。

【請求項6】 前記第2のユーザによる前記第1のサブセットへのアクセスのための前記第2の要求にตอบสนองして、前記第1のサブセットへの書き込みロックを探索するステップと、

前記第1のサブセットへの書き込みロックを見出したことにตอบสนองして、予約キュー内に要求オブジェクトを配置するステップと、

を含む、請求項1記載の方法。

【請求項7】 前記第1のサブセットへの書き込みロックが、前記第1のサブセットを記述するロック・キュー内のロック・オブジェクトである、請求項1記載の方法。

【請求項8】 コンピュータ・システムのメモリ内の複数のサブセットから成るデータ・セットへの複数のユーザによる同時アクセスを制御するシステムであって、データ・セットの第1のサブセットへの書き込みロックを探索する手段と、

第1のユーザによる前記第1のサブセットへの書き込みアクセスのための第1の要求を許可する手段と、

他のユーザが前記第1のサブセットを変更することを禁止するために、前記データ・セットの前記第1のサブセ

ットに書き込みロックを設定し、他のユーザによる前記第1のサブセットへの読出しアクセスは許可する手段と、

を含むシステム。

【請求項9】 前記データ・セットの第2のサブセットへの書き込みロックを探索する手段と、

第2のユーザによる前記第2のサブセットへの書き込みアクセスのための第2の要求を許可する手段と、

他のユーザが前記第2のサブセットを変更することを禁止するために、前記データ・セットの前記第2のサブセ

ットに書き込みロックを設定し、他のユーザによる前記第2のサブセットへの読出しアクセスは許可する手段と、

を含む、請求項8記載のシステム。

【請求項10】 前記第1のサブセット内の要素の第1のセット、及び前記第1のサブセット外の要素の第2のセ

ットを有する、前記データ・セットの第3のサブセ

ットへの書き込みロックを探索する手段と、

第3のユーザによる前記第3のサブセットの要素の前記第2のセットへの書き込みアクセスのための第3の要求を許可する手段と、

前記第3のサブセットの要素の前記第2のセットへの書き込みロックを設定する手段と、

を含む、請求項8記載のシステム。

【請求項11】 前記第1のユーザからのコマンドにตอบสนองして、前記第1のサブセットを変更する手段と、

前記第1のサブセット内の変更を反映するために、前記データ・セットを更新する手段と、

50

前記第1のサブセットへの書き込みロックを解除する手段と、

を含む、請求項8記載のシステム。

【請求項12】前記第1のサブセットへの書き込みロックが通知モードであり、

前記第2のユーザによる前記第1のサブセットへの書き込みアクセスのための前記第2の要求に回答して、前記第1のサブセットへの書き込みロックを探索する手段と、

前記第1のユーザに、前記第1のサブセットに対する第2の要求が受信されたことを示すメッセージを送信する手段と、

所定期間内に前記第1のユーザによる応答が受信されない場合、前記第1のサブセットへの書き込みロックを解除する手段と、

を含む、請求項8記載のシステム。

【請求項13】前記第2のユーザによる前記第1のサブセットへの書き込みアクセスのための第2の要求に回答して、前記第1のサブセットへの書き込みロックを探索する手段と、

前記第1のサブセットへの書き込みロックを見い出したことに回答して、予約キュー内に要求オブジェクトを配置する手段と、

を含む、請求項8記載のシステム。

【請求項14】前記第1のサブセットへの書き込みロックが、前記第1のサブセットを記述するロック・キュー内のロック・オブジェクトである、請求項8記載のシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はコンピュータ・システムにおけるデータ記憶装置に関し、特に、バージョン化オブジェクトのロッキング方法及びシステムに関する。

【0002】

【従来の技術】複数のプロセッサがネットワーク内において相互接続される分散コンピュータ環境が普及しつつある。これらの環境では、何人かのユーザが同時に同一のバージョン化オブジェクトにアクセス及び潜在的には更新することを望む可能性がある。バージョン化オブジェクトは、複数のユーザに対する複数のコピーを有するオブジェクトである。バージョン制御は、例えばソフトウェア・エンジニアリング、CAD/CAMシステム、及び複合文書を扱うオフィス情報システムなどの様々なデータ集中型アプリケーション分野における重要な機能の1つである。バージョン制御は、データの単一のバージョンが全てのユーザによって使用可能であることを保証する。更に、これらの環境のユーザは、オブジェクトの複数のバージョンを試してみる必要がある。

【0003】従来の読出し/書き込みロッキング機構は親オブジェクトであるか子オブジェクトかを区別せず、また部分オブジェクトを扱わない。ユーザは全オブジェ

クトをロックまたはアンロックしなければならない。更に、誰かがオブジェクトをロックしたことによりロック要求が失敗する場合、ユーザはオブジェクトが使用可能になるまで、何度もロック要求を再試行しなければならない。

【0004】従来の方法によれば、トークンの使用を通じて、複数のユーザがデータ・ファイルへのアクセスを共有することができる。複数のユーザはデータの1部に対する読出しトークンを有するが、あるユーザが読出し/書き込みトークンを獲得する時、他のユーザにより保持される全ての読出しトークンは無効となる。読出し/書き込みトークンを有するユーザは、ユーザが変更を希望するファイルの論理部分だけを制御するのではなく、そのファイルのために使用される記憶装置の物理ページを制御する。

【0005】

【発明が解決しようとする課題】本発明の目的は、分散データ処理環境において、複数のユーザによるバージョン化オブジェクトへの同時アクセスを支援することである。

【0006】本発明の別の目的は、バージョン化オブジェクトの1部に対するあるユーザの変更にもとづき、バージョン化オブジェクトの複数のコピーを更新することである。

【0007】本発明の更に別の目的は、異なるユーザによる排他的書き込みアクセスに対し、オブジェクトの異なる論理部分をロックすることである。

【0008】

【課題を解決するための手段】これらの目的が、バージョン化オブジェクトの同時制御を扱うために設計される複数のロック・モードにより達成される。"協調 (Cooperative)"、"通知 (Notify)"及び"予約 (Reserve)"ロック・モードは、オブジェクトの同時使用を増加するために設計される特定のロック・モードである。

【0009】複数のユーザによるコンピュータ・システムの共用メモリへの同時アクセスのための方法、システム及びプログラムは、最初に、ユーザがアクセスを希望するデータ・セットの第1の部分に対するロックを探索する。ロックが検出されない場合、所望の部分に対するアクセスを許可することにより、要求が許可される。それから、ロックがデータ・セットの所望の部分に設定され、他のユーザがアクセスすることを禁止する。第2のユーザがデータ・セットの第2の部分の希望する場合についても、処理は同様である。データ・セットの第2の部分に対するロックが探索される。ロックが見い出されない場合、第2の部分へのアクセスのための第2のユーザの要求が許可される。他のユーザのアクセスを禁止するために、第2の部分に対するロックが設定される。第3のユーザがデータ・セットの第3の部分へのアクセスを希望し、第3のデータ・セットが、既にロックが設定

された第1の部分と幾つかの共通要素を共有する場合、第3のユーザの要求は部分的に許可される。彼は、第1の部分により共有されない第3の部分の1部に対する排他的書き込みロックを受信する。ユーザはオブジェクトの任意部分に対する排他的書き込みロックの有無に関わらず、一般にオブジェクトの全ての部分に対する読出しアクセスを有する。ユーザがデータ・セットの自身の部分に関して終了すると、ロックが除去され、ユーザの更新部分のレコード数が、最初に受信された時のこの部分のレコード数と比較され、データ・セットの親コピーがそれに従い更新される。

【0010】通常、ユーザがデータ・セットの予め予約された部分へのアクセスを要求する時、要求は拒絶される。しかしながら、本発明はこうした状況を緩和する2つのモードを含む。ユーザが実際にデータ・セットの1部を必要かどうかの不確かであるが、将来の使用のために所望の部分にロックを設定したい場合、ユーザは通知(Notify)モードによりロックする。第2のユーザによる第2の要求が検出されると、通知ロックを有するユーザに対し、所望部分に対する第2の要求が受信されたことを示すメッセージが送信される。ユーザはロック・モードを排他的結合アクセス・モードに変更するか、何もアクションを起こさない場合には、所望部分に対する彼のロックが除去される。予約(Reserve)モードは、ロックの探索により所望部分が既に使用されていることが示される場合に、所望部分に対してユーザが予め順番待ちすることを許可する。

【0011】

【実施例】本発明は多数の異なるオペレーティング・システム下で動作する様々なコンピュータまたはコンピュータの集合において実行される。コンピュータは例えばパーソナル・コンピュータ、ミニ・コンピュータ、またはメインフレーム・コンピュータなどである。ローカル・エリア・ネットワークまたはワイド・エリア・ネットワークまたは大規模テレプロセッシング・システムなどのネットワークの1部である複数のコンピュータが使用される。コンピュータの特定の選択はディスク及びディスク記憶要求によってのみ制限されるが、IBM PS/2 (TM) シリーズのコンピュータが本発明においては使用される。IBMのPS/2シリーズ・コンピュータに関する詳しい情報については、Technical Reference Manual Personal System/2 Model 50, 60 Systems IBM Corporation, Part No. 68X2224 Order Number S68X-2224 及び Technical Reference Manual Personal Systems/2 (model 80) IBM Corporation Part No. 68X2256 Order Number S68X-2254 等を参照されたい。IBM PS/2 パーソナル・コンピュータが動作するオペレーティング・システムの1つは、IBMのOS/2 2.0 (TM) であり、IBM OS/2 2.0 オペレーティング・システムに関する詳しい情報については、OS/2 2.0

Technical Library, Programming Guide Vol. 1, 2, 3 Version 2, 00 Order Nos. 10G6261, 10G6495, 10G6494 等を参照されたい。

【0012】また、コンピュータ・システムがAIX (TM) オペレーティング・システム上で動作するIBM RISC System/6000 (TM) ラインのコンピュータであってもよい。RISC System/6000の様々なモデルがIBMからの多くの刊行物に述べられており、例えば、RISC System/6000, 7073 and 7016 POWERstation and POWERserver Hardware Technical reference, Order No. SA23-2644-00などがある。AIXオペレーティング・システムについては、General Concepts and Procedure—AIX Version 3 for RISC System/6000 Order No. SC23-2202-00及びIBMからの他の刊行物に述べられている。

【0013】図1において、システム・ユニット11、キーボード12、マウス13、及び表示装置14を含むコンピュータ10が示されている。表示装置14の画面16はデータ・オブジェクトに対する可視的变化を提供するために使用される。オペレーティング・システムにより支援されるグラフィカル・ユーザ・インタフェースは、ユーザがポインタ15を画面16上の特定の位置に位置するデータ・オブジェクトを表すアイコンに移動することにより、入力に関するポイント及びシュート方法の使用を可能とし、またマウス・ボタンの1つを押してユーザ・コマンドまたは選択を実行することを可能とする。

【0014】図2は図1に示されるパーソナル・コンピュータの構成要素のブロック図を示す。システム・ユニット11はシステム・バスまたは複数のシステム・バス21を含み、これに様々な構成要素が接続され、それにより様々な構成要素間の通信が達成される。マイクロプロセッサ22はシステム・バス21に接続され、同様にシステム・バス21に接続される読出し専用メモリ(ROM)23及びランダム・アクセス・メモリ(RAM)24により支援される。IBMマルチメディアPS/2シリーズのコンピュータ内で使用されるマイクロプロセッサは、インテル社のマイクロプロセッサ・ファミリの1つであり、これには386または486マイクロプロセッサが含まれる。しかしながら、他のマイクロプロセッサも使用可能であり、例えばモトローラ社のマイクロプロセッサ・ファミリに含まれる68000, 68020または68030マイクロプロセッサ、及びIBM、ヒューレット・パカード、サン、インテル、モトローラ、及びその他による様々な縮小命令セット・コンピュータ(RISC)・マイクロプロセッサなどが特定のコンピュータにおいて使用される。

【0015】ROM23は他のコードと共に基本入出力システム(BIOS)を含み、これは対話及びディスク・ドライブ及びキーボードなどの基本ハードウェア動作

を制御する。RAM24は主メモリであり、これにはオペレーティング・システム及びアプリケーション・プログラムがロードされる。メモリ管理チップ25がシステム・バス21に接続され、RAM24とハード・ディスク・ドライブ26及びフロッピー・ディスク・ドライブ27間のデータ転送などの直接メモリ・アクセス動作を制御する。CD-ROMドライブ32がまたシステム・バス21に接続され、例えばマルチメディア・プログラムまたはプレゼンテーションなどの大量のデータを記憶するために使用される。

【0016】システム・バス21には更に様々な入出力制御装置が接続される。これらにはキーボード制御装置28、マウス制御装置29、ビデオ制御装置30、及び音声制御装置31が含まれる。キーボード制御装置28はキーボード12に対応するハードウェア・インタフェースを提供し、マウス制御装置29はマウス13に対応するハードウェア・インタフェースを提供し、ビデオ制御装置30は表示装置14に対応するハードウェア・インタフェースである。入出力制御装置40は例えばトークン・リング・アダプタであり、ネットワーク42を介して、他の類似に構成されるデータ処理システムとの通信を可能とする。

【0017】メモリ24内の複数のコード・モジュール内で実現されるバージョン制御システムの構成が図3に示される。

【0018】バージョン制御システム (version control system) は、RAM24内に活動状態で存在しない時には、フロッピー・ディスク・ドライブ用のフロッピー・ディスク、CD-ROMドライブ用のCD-ROMなどのコンピュータ読み出し可能媒体上、またはハード・ディスク・ドライブ上に記憶される。

【0019】バージョン制御システムは次の主要構成要素を含む。バージョン・マネージャ (version manager) 100、ロック・マネージャ (lock manager) 101、及び固有ファイル・システム (native file system) 102。バージョン・マネージャ100は、ユーザからチェック・イン (checkin) 及びチェック・アウト (checkout) ・コマンドを獲得するために、ユーザと対話する。バージョン・マネージャ100は値が適切であることを確認するために、パラメータを解析する。次に、バージョン・マネージャ100はLockParentコマンドをロック・マネージャ101に発行し、ユーザが子オブジェクトを生成可能かどうかをチェックする。ロック・マネージャ101は全てのチェック・イン及びチェック・アウト・レコードを保持する。ロック・マネージャ101はチェック・アウト・キュー103及び予約済みのチェック・アウト・キュー104を保持し、オブジェクトの状態を追跡する。チェック・アウト・キュー103は、部分または全体オブジェクトをチェック・アウトしたユーザを追跡する。予約済みチェック・アウト・キ

ュー104は、オブジェクトへのアクセスを予約したユーザを追跡する。

【0020】ロック・マネージャ101がロック・アクセスを承認した後、バージョン・マネージャ100は固有ファイル・システム102を呼出し、親オブジェクトから子オブジェクトを生成する。子オブジェクト識別子がユーザに返却される。

【0021】ユーザはオブジェクトをバージョン化オブジェクトまたは非バージョン化オブジェクトとして宣言することができる。ユーザはバージョン化オブジェクトにアクセスするために、バージョン・マネージャにより提供される "チェック・アウト" ・コマンド及び "チェック・イン" ・コマンドを使用する。新たなバージョンは既存のバージョンの更新時に、その既存のバージョンから生成される。バージョン・マネージャ100はバージョン派生階層を追跡する。

【0022】ユーザがバージョン化オブジェクトをチェック・アウトするために "チェック・アウト" ・コマンドを使用する時、バージョン・マネージャはユーザ用の一時的なバージョンを生成する。全ての更新がこの一時オブジェクトに適応される。この一時オブジェクトはこのユーザにのみ属するために、彼はロック・マネージャ100により、自動的に一時オブジェクトに対する排他的ロックを獲得する。

【0023】同一の親バージョンは、通常、他のユーザによりチェック・アウトされるため、一時オブジェクトが存在する時に、親オブジェクトへのアクセスが制御されなければならない。ロッキング・コマンドは "lockParent" と称され、新たなバージョンが同一の親バージョンから生成されるかどうかを制御する。このコマンドはバージョン・マネージャ100により使用される。ユーザはチェック・アウト・コマンドを発行する時、ロック・モード及びフラグも提示しなければならない。次にバージョン・マネージャ100が、ユーザにより提供される情報にもとづき、LockParentを発行する。

【0024】LockParentコマンドの形式を次に示す。

LockParent (userid, objectId, mode, from, to)

【0025】"userid" はユーザ識別子 (またはログオン・ユーザid) を示す。"objectId" は親オブジェクトのオブジェクト識別子を示す。"mode" はロック・マネージャにより提供されるアクセス・モードを示す。"from" 及び "to" はユーザがチェック・アウトを希望する開始レコード及び終了レコードを示す。

【0026】ロック・マネージャは次のアクセス・モードを提供する。

【0027】"R" は要求される親オブジェクトに対する "参照 (reference)" アクセスを提供する。親バージョンが "参照" モードにロックされている場合には、他のユーザはこの親バージョンを依然としてチェック・アウトでき、チェック・アウト・コマンドを通じ、並列バー

ジョンを生成できる。

【0028】"F"は要求される親オブジェクトに対する"フリーズ(freeze)"アクセスを提供する。このアクセスが許可されると、どのユーザもこのオブジェクトから永久バージョンを生成することが許可されない。しかしながら、参照モードにおけるロッキング要求に対する一時バージョンは生成される。

【0029】"C"は要求される親オブジェクトに対する"協調(cooperative)"アクセスを提供する。このモードでは、要求される親オブジェクトから並列な子バージョンが生成される。"from"及び"to"レコード番号を提供することにより、ユーザは部分オブジェクトをチェック・アウトできる。ロック・マネージャは、各ユーザの作業が衝突しないことを保証し、オブジェクトが再度チェック・インされる時に、非衝突作業を併合する。實際上、各ユーザは親オブジェクトの特定部分への"フリーズ"・アクセス、及び全ての他の部分への"参照"アクセスを有する。

【0030】"N"は要求される親オブジェクトに対する"通知(notify)"アクセスを提供する。このロック・モードは、チェック・アウトしたオブジェクトをどのように処置するか定まっていないうーザのためのモードである。ロック・マネージャは彼らに判断を延期させる。

【0031】"RE"は親オブジェクトに対する"予約(reserve)"アクセスを提供する。ユーザはバージョン化オブジェクトへのアクセスを予約することができる。

【0032】参照モード("R")は、アイデアを借りるために関連ソース・オブジェクトを見ようとするプログラマにとって、特に有用である。この場合、彼は別の人間がそのオブジェクトの新たなバージョンを生成しているかどうかを全く気にしない。"チェック・イン"・コマンドは"参照"モードでは禁止される。協調モード("C")は、複数のユーザが同一のソース・ファイル(但し異なるモジュール)を処理する時に有用である。協調アクセス・モードでは、複数の人間が同一の親バージョンから子バージョンを生成することができる。

【0033】通知モード("N")は、誰か別の人間が同一の親オブジェクトに対する"フリーズ"・ロックを要求するまで、ユーザが実際のロック・モードに関する決定を延期することを許可する。この機能は従来のユーザによる不要なロック要求を低減し、同時オブジェクト使用を増加させる。

【0034】予約モード("RE")は、ユーザが特定のオブジェクトをアクセスするための予約を許可する。通常、親オブジェクトが幾人かの他のユーザにより"フリーズ"・モードでロックされる場合、ユーザはオブジェクトが使用可能となるまで、試行を継続しなければならない。

【0035】バージョン制御システムの協調アクセス・モードが、図4乃至図6に示される。

【0036】ソフトウェア開発環境においては、複数の人間が同一のソース・オブジェクト(但しそのソース・オブジェクトの異なる部分)に対し作業する必要があり、これは特に大規模なソース・オブジェクトにおいてその可能性が高い。協調アクセス・モードでは、ユーザは自身が関心のあるオブジェクトの1部だけをチェック・アウトすなわちフリーズすることができる。ロック・マネージャはユーザの作業が他のユーザと衝突しないことを保証し、オブジェクトがチェック・インされる時に、衝突しない作業を併合する。

【0037】図4を参照すると、処理はステップ150で開始され、オブジェクトをチェック・アウトするユーザ要求を受信する。次にステップ152で、ロック・マネージャはユーザ要求を許可するか否かを判断する。ユーザ要求が以前に許可された要求と衝突する場合、ロック・マネージャは要求を拒否し、処理はステップ154で終了する。

【0038】上述のように、協調アクセス・モードを支援する場合、ロック・マネージャは部分オブジェクトまたは全体オブジェクトをチェック・アウトしたユーザを追跡するために、チェック・アウト・キューを維持する。要求が許可されると、ステップ156でロック・オブジェクトがチェック・アウト・キューに追加される。

【0039】ロック・オブジェクトは次の情報を有する。

(1) ユーザID: ロックを要求するユーザの名前。

(2) From: 開始レコード番号。LockParentコマンド内に"from"パラメータが指定されない場合、"From"は0にセットされる。

(3) To: 終了レコード番号。LockParentコマンド内に"to"パラメータが指定されない場合、"to"は最大値にセットされる。

(4) TranObjectId: 一時オブジェクト識別子。

(5) ロック・モード: 上述した通りであり、以降で更に詳述される。

(6) Timestamp: 要求時のタイムスタンプ(日時)。

(7) 優先順位: ロック要求の優先順位。

【0040】ロック・マネージャはステップ150で要求を受信すると、最初にユーザが要求したレコード範囲が、以前に許可された既存のLockParent要求のレコード範囲と衝突するかどうかをチェックする。この処理は図5に、より詳細に示される。

【0041】ユーザはオブジェクトの更新を終了後、ステップ158で、チェック・イン・コマンドにより部分オブジェクトをチェック・インする。ステップ160で、バージョン・マネージャはロック・マネージャにUnLockParentコマンドを発行する。ステップ162で、ロック・マネージャは部分オブジェクト識別子(PartialObjectId)を使用することにより、ロック・キューから正しいロック・オブジェクトを識別する。次にステップ

164で、ロック・マネージャは部分オブジェクトの新旧バージョン内のレコード数の差から、新たな部分オブジェクトに対応するレコード数が変化したかどうかをチェックする。ユーザがレコードを追加または消去していない場合（すなわち差が0の場合）、ステップ166で、ロック・マネージャはバージョン・マネージャに、“from”から“to”までのレコードを置換することにより、親オブジェクトの新たなバージョンを生成するように依頼する。それ以外の場合、ロック・マネージャはステップ168で、ロック・キュー内の全てのロック・オブジェクトのfrom及びto情報を更新する。これにより親オブジェクトが更新される。

【0042】ロック・マネージャがオブジェクトのチェック・アウト要求が許可されるべきかどうかを決定する処理が図5に示される。ステップ180で、ロック・キュー内の各ロック・オブジェクトに対応して、ロック・キューがオブジェクトID (objectId) により検索される。ステップ182で、ロック・マネージャはLockParent要求により指定されるレコード範囲が、ロック・キュー内の各ロック・オブジェクトに対してテストされることを保証する。ステップ184でロック・キュー内にロ*

procedure LockParent (userid, objectId, from, to, mode)

{
 ロック・キュー内の各ロック・オブジェクトLにおいて、idにobjectIdを
 有するオブジェクトのロック・キューを獲得

{
 /*間隔[from, to]が間隔[L, from, L, to]に重なるかをチェック*/
 if L, from < from < L, to
 thenロック要求を拒否
 else
 if L, from < to < L, to
 thenロック要求を拒否
 else
 if from < L, from < to
 thenロック要求を拒否
 else
 if from < L, to < to
 thenロック要求を拒否
 else
 ロック要求を許可し、ロック要求をロック・キュー内に配置する

}

【0044】図6は、変更された部分オブジェクトが親オブジェクト及びロック・キューを更新するために使用される様子を表し、この処理はロック・マネージャにより実行される。ステップ200で、部分オブジェクトに対応するロック・オブジェクトが検索される。次にステップ202で、部分オブジェクトのレコード数が検索される。ステップ204で、部分オブジェクトのオリジナル・レコード数が検索される。ステップ206で、レコ

*ック・オブジェクトが存在しない場合、ロック要求が部分オブジェクトに対して許可される。ステップ186乃至ステップ192では、現LockParent要求のレコード範囲が、現在テストされているロック・オブジェクトLのレコード範囲内に存在するかどうかを判断するための、一連のテストが実行される。ステップ186で、現要求の開始レコードすなわち“from”が現ロック・オブジェクトの範囲内に存在する場合、要求内の幾つかのレコードがロック・オブジェクト内のレコードをオーバーラップするため、要求がステップ194で拒否される。同様に、終了レコード例えば現要求の“to”が、ロック・オブジェクトの範囲内すなわちL, from 乃至L, to の範囲内に存在する場合、要求は拒否される。ステップ190及び192では、ロック・オブジェクトの開始及び終了のいずれかが現要求の範囲内に存在するかどうかテストされる。現要求内のレコードがロック・キュー内のロック・オブジェクトにより以前にロックされていない場合、要求がステップ184で許可される。

【0043】次にLockParent要求のコード・サンプルを示す。

ードの現在数とオリジナル数との差が計算される。次にステップ208で、差が0に等しいかどうかテストされ、等しい場合、オリジナル部分オブジェクト内のレコード数と同数のレコードが、更新される部分オブジェクト内に存在する。レコード数が変更されない場合、ステップ210で親オブジェクトが更新され、ステップ220でロック・オブジェクトがロック・キューから消去され、処理は終了する。差が0に等しくない場合、レコ

13

ド数が変更されたことになる。従って、ステップ222で、各ロック・オブジェクトがロック・キューから検索され、更新される部分オブジェクトよりも大きなレコード数を有するロック・オブジェクトについては、それらの開始パラメータ及び終了パラメータが増減されたり、*

```
procedure UnlockParent (userid, objectId, partialObjectId)
```

```
{
```

```
    partialObjectIdを有するロック・オブジェクト (L) を獲得
```

```
    partialObjectのレコード数R1を獲得
```

```
    partialObjectのオリジナル・レコード数R2を計算
```

```
    ここでR2 = (L. to - L. from) + 1
```

```
    差D = R1 - R2を計算
```

```
    if D = 0
```

```
    then
```

```
    {
```

```
        親オブジェクトを新たなデータで更新
```

```
        ロック・キューからLを消去
```

```
    }
```

```
    else
```

```
    {
```

```
        キュー内の各ロック・オブジェクトL1に対応して、idにobjectIdを有するオブジェクトのロック・キューを獲得
```

```
        {
```

```
            if (L1. from > L. from) or (L1. to > L. to)
```

```
            then
```

```
            {
```

```
                L1. from = L1. from + D
```

```
                L1. to = L1. to + D
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

【0046】図7及び図8は、LockParent及びUnlockParent手順を介して、同時アクセスが提供されるソース・オブジェクトを示す。この例のソース・オブジェクト名は"Part. c"である。図7においてブロック300として示される"Part. c"オブジェクトの1部をチェック・アウトした2人のユーザが現在存在する。トム (Tom) はライン100N乃至200Nまでのソース・コードをチェック・アウトしている。マイク (Mike) はライン300N乃至350Nまでのソース・コードをチェック・アウトしている。オブジェクト300のチェック・アウトされた部分コピー303、305が、ロック・キュー310内のロック・オブジェクト306及び308により反映されている。ピーター (Peter) がPart. cをライン150N乃至250Nまでチェック・アウトしようとする、彼の要求は拒否される。なぜなら、150N乃至200NまでがTomによりチェック・アウトされているからである。Part. cをライン360N乃至400Nまでチェック・アウトするPeterの要求は、バージョン

14

*更新される部分オブジェクトとオリジナル部分オブジェクトとの差に従い調整される。

【0045】次にUnlockParentコマンドのサンプル・コード例を示す。

・マネージャにより許可される。なぜなら、これらのレコードに対しては、既存のロック・オブジェクトが存在しないからである。

【0047】図8では、Tomは彼の作業を終え、彼の部分オブジェクト303をチェック・インしようとする。彼はチェック・アウトしたいいくつかのソース・コードを消去したため、彼の部分オブジェクトは61ラインだけを有する (元々は101ライン)。バージョン制御システムはキュー上のロック要求を更新する。結果が図8に示される。Mikeが彼の部分オブジェクト305をチェック・インする時、彼の開始ライン番号は300Nの代わりに260Nとなる。

【0048】通知ロック・モードは、チェック・アウトしたオブジェクトをどのように処理しようか定まていないユーザのためのモードである。ロック・マネージャは、誰か別のユーザが同一の親オブジェクトに対する"フリーズ"・ロックを要求するまで、ユーザが判断を延期することを許可する。この手順が図9に示される。

【0049】ステップ330でロック・マネージャが通知ロック・モードのLockParent要求を受信すると、これは最初にユーザが要求したレコードの範囲が、以前のLockParent要求からの既存のロックと衝突するかどうかをチェックする。衝突する場合、LockParent要求は拒否され、処理はステップ334で終了する。それ以外の場合、LockParent要求が許可され、ステップ336でロック・オブジェクトがチェック・アウト・キューに追加される。ロック・オブジェクトのモード・フィールドがステップ338で通知ロック・モードにセットされる。

【0050】ステップ340でロック・マネージャが第2のLockParent要求が既存のロックと衝突することを見出し、親オブジェクトが通知ロック・モードでチェック・アウトされた場合、次の処理が実行される。

【0051】ステップ342では、通知 (note) がロック・ホルダのリーダに送信され、ロック・ホルダにロック・モードを変更するように依頼する。次にステップ344で、ロック・モードが1日以内で変更されない場合、ロック・モードが“参照”に変更され、“フリーズ”・ロックを要求したユーザが親オブジェクトをフリーズする。

【0052】通常、用心深いユーザは、変更を行うかどうか不確かな場合、オブジェクトをチェック・アウトする傾向がある。従来技術では本発明の“フリーズ”・モードに類似のロックを含むだけなので、他のユーザがオブジェクトをアクセスすることを禁止した。用心深いユーザが長い期間オブジェクトをチェック・アウトすることは不便なことである。“通知”モードは従来のユーザによる幾つかの不要なロック要求を低減し、同時オブジェクト使用を増加する。

【0053】予約ロック・モードの目的は、オブジェクトが使用可能でない時、ユーザが繰返しオブジェクトのチェック・アウトを試行することなく、ユーザがソース・オブジェクトのアクセスの予約をできるようにする。この処理を図10に示す。

【0054】ロック・マネージャは要求を受信した後、最初にステップ352で、オブジェクトがチェック・アウトされたかどうかをチェックする。オブジェクトがチェック・アウトされていない場合、オブジェクトは排他的アクセス・モードによりロックされ、許可オブジェクトがユーザに返却される。フローはステップ156から開始され、図4乃至図6に示されるように継続する。オブジェクトがチェック・アウトされている場合、ロック・マネージャはステップ356で要求オブジェクトを予約キューに配置する。

【0055】ステップ358では、通知 (note) がユーザに送信され、彼の要求が予約キューに配置されたことを通知する。同時にステップ360で、ロック・マネージャはバージョン・マネージャに、現在オブジェクトを保持しているユーザに助言を送信するように通知する。

ユーザがオブジェクトをチェック・インするまで、24時間毎に通知が送信される。

【0056】ステップ362でオブジェクトがチェック・インされると、ロック・マネージャはステップ364で予約キューをチェックし、最も高い優先順位の要求オブジェクトを取り出す。同一の優先順位の複数の要求オブジェクトが存在する場合には、最小のタイムスタンプのオブジェクトが選択される。

【0057】ステップ366では、要求オブジェクト内にユーザIDが存在するユーザに通知 (note) が送信される。ユーザはステップ368で、24時間以内または他の所定期間以内にオブジェクトをチェック・アウトする必要がある。もしそのようにすると、ステップ370により、図4乃至図6に示される通常の処理が継続される。それ以外の場合、システムはステップ372で予約キューから第2の要求オブジェクトを選択する。

【0058】ステップ374でオブジェクトをチェック・アウトする人間が、ステップ376でそのオブジェクトを消去するように決定することが可能である。この場合、ステップ378で予約キュー内の全ての現ユーザに、オブジェクトが消去され、キュー内の全ての予約オブジェクトが消去されることが通知される。

【0059】図11は予約処理を表す。予約キュー400内にオブジェクト410内のSample.cに対応する3つの要求オブジェクトA、B、Cが存在する。システムが“Checkin”コマンドを受信すると、要求オブジェクトAが選択される。なぜなら、これは最も高い優先順位及び最小のタイムスタンプを有するからである。ユーザTomにモジュールをチェック・アウトするように依頼する通知 (note) が送信される。Tomは1日以内または他の所定期間以内にSample.c400をチェック・アウトする必要がある。そうしない場合、システムは、要求オブジェクトCと同一の優先順位ではあるが、より小さなタイムスタンプを有する要求オブジェクトBを保持するPeterに通知 (note) を送信する。

【0060】下記の表は要求モードの互換性マトリクスを提供する。行は要求されるロック・モードに対応し、列は別のトランザクションにより既に保持されているロック・モード、すなわちロック・キュー内のエントリのロック・モードに対応する。可能なロック要求には、R (referenced)、F (freeze)、N (notify)、C (corporate)、RE (reserve) がある。NLはどのトランザクションも、親オブジェクトに対するロックを保持していないことを意味する。エントリが“Yes”の場合、要求されるロックが許可され、“No”の場合には許可されない。他のエントリには要求の不在を表す“NL”、ロックを要求したユーザにロックの現状態が通知される“NFree”、現ロック・ホルダに他の人間が丁度このオブジェクトへのアクセスを要求していることが通知される“NFlow”、及びロック要求が予約キュー内に配置されること

を意味する"Rev"が含まれる。

【0061】

【表1】

	R	F	N	C	RE
R	Yes	Yes	Yes	Yes	Yes
F	Yes	No	NfRe	No	No
N	Yes	NfOwn	NfOwn	NfOwn	NfOwn
C	Yes	No	NfRe	Yes	No
NL	Yes	Yes	Yes	Yes	Yes
RE	Rev	Rev	NfOwn, Rev	Rev	Rev

【0062】以下の通りに発明を開示する。

(1) コンピュータ・システムのメモリ内の複数のサブセットから成るデータ・セットへの複数のユーザによる同時アクセスを制御する方法であって、前記データ・セットの第1のサブセットへの書込みロックを探索するステップと、第1のユーザによる前記第1のサブセットへの書込みアクセスのための第1の要求を許可するステップと、他のユーザが前記第1のサブセットを変更することを禁止するために、前記データ・セットの前記第1のサブセットに書込みロックを設定し、他のユーザによる前記第1のサブセットへの読出しアクセスは許可するステップと、を含む方法。

(2) 前記データ・セットの第2のサブセットへの書込みロックを探索するステップと、第2のユーザによる前記第2のサブセットへの書込みアクセスのための第2の要求を許可するステップと、他のユーザが前記第2のサブセットを変更することを禁止するために、前記データ・セットの前記第2のサブセットに書込みロックを設定し、他のユーザによる前記第2のサブセットへの読出しアクセスは許可するステップと、を含む、(1)記載の方法。

(3) 前記第1のサブセット内の要素の第1のセット、及び第1のサブセット外の要素の第2のセットを有する、前記データ・セットの第3のサブセットへのロックを探索するステップと、第3のユーザによる前記第3のサブセットの要素の前記第2のセットへの書込みアクセスのための第3の要求を許可するステップと、前記第3のサブセットの要素の第2のセットへの書込みロックを設定するステップと、を含む、(1)記載の方法。

(4) 前記第1のユーザからのコマンドにตอบสนองして、前記第1のサブセットを変更するステップと、前記第1のサブセット内の変更を反映するために、前記データ・セットを更新するステップと、前記第1のサブセットへの書込みロックを解除するステップと、を含む、(1)記載の方法。

(5) 前記第1のサブセットへの書込みロックが通知モードであり、前記第2のユーザによる前記第1のサブセットへの書込みアクセスのための前記第2の要求にตอบสนองして、前記第1のサブセットへの書込みロックを探索するステップと、前記第1のユーザに、前記第1のサブセ

ットに対する前記第2の要求が受信されたことを示すメッセージを送信するステップと、所定期間内に前記第1のユーザによる応答が受信されない場合、前記第1のサブセットへの書込みロックを解除するステップと、を含む、(1)記載の方法。

(6) 前記第2のユーザによる前記第1のサブセットへのアクセスのための前記第2の要求にตอบสนองして、前記第1のサブセットへの書込みロックを探索するステップと、前記第1のサブセットへの書込みロックを見い出したことにตอบสนองして、予約キュー内に要求オブジェクトを配置するステップと、を含む、(1)記載の方法。

(7) 前記第1のサブセットへの書込みロックが、前記第1のサブセットを記述するロック・キュー内のロック・オブジェクトである、(1)記載の方法。

(8) コンピュータ・システムのメモリ内の複数のサブセットから成るデータ・セットへの複数のユーザによる同時アクセスを制御するシステムであって、データ・セットの第1のサブセットへの書込みロックを探索する手段と、第1のユーザによる前記第1のサブセットへの書込みアクセスのための第1の要求を許可する手段と、他のユーザが前記第1のサブセットを変更することを禁止するために、前記データ・セットの前記第1のサブセットに書込みロックを設定し、他のユーザによる前記第1のサブセットへの読出しアクセスは許可する手段と、を含むシステム。

(9) 前記データ・セットの第2のサブセットへの書込みロックを探索する手段と、第2のユーザによる前記第2のサブセットへの書込みアクセスのための第2の要求を許可する手段と、他のユーザが前記第2のサブセットを変更することを禁止するために、前記データ・セットの前記第2のサブセットに書込みロックを設定し、他のユーザによる前記第2のサブセットへの読出しアクセスは許可する手段と、を含む、(8)記載のシステム。

(10) 前記第1のサブセット内の要素の第1のセット、及び前記第1のサブセット外の要素の第2のセットを有する、前記データ・セットの第3のサブセットへの書込みロックを探索する手段と、第3のユーザによる前記第3のサブセットの要素の前記第2のセットへの書込みアクセスのための第3の要求を許可する手段と、前記第3のサブセットの要素の前記第2のセットへの書込みロックを設定する手段と、を含む、(8)記載のシステム。

(11) 前記第1のユーザからのコマンドにตอบสนองして、前記第1のサブセットを変更する手段と、前記第1のサブセット内の変更を反映するために、前記データ・セットを更新する手段と、前記第1のサブセットへの書込みロックを解除する手段と、を含む、(8)記載のシステム。

(12) 前記第1のサブセットへの書込みロックが通知モードであり、前記第2のユーザによる前記第1のサブ

10

20

30

40

50

セットへの書込みアクセスのための前記第2の要求に
 10 答して、前記第1のサブセットへの書込みロックを探索
 する手段と、前記第1のユーザに、前記第1のサブセ
 ットに対する第2の要求が受信されたことを示すメッセ
 ージを送信する手段と、所定期間内に前記第1のユーザに
 による応答が受信されない場合、前記第1のサブセットへ
 の書込みロックを解除する手段と、を含む、(8)記載
 のシステム。

(13) 前記第2のユーザによる前記第1のサブセット
 への書込みアクセスのための第2の要求に答して、前
 記第1のサブセットへの書込みロックを探索する手段
 と、前記第1のサブセットへの書込みロックを見出し
 たことに答して、予約キュー内に要求オブジェクトを
 配置する手段と、を含む、(8)記載のシステム。

(14) 前記第1のサブセットへの書込みロックが、前
 記第1のサブセットを記述するロック・キュー内のロッ
 ク・オブジェクトである、(8)記載のシステム。

(15) コンピュータ・システムのメモリ内のデータ・
 セットへの複数のユーザによる同時アクセスを制御する
 コンピュータ読出し可能媒体上のコンピュータ・プログ
 ラム・プロダクトであって、データ・セットの第1のサ
 20 ブセットへの書込みロックを探索する手段と、第1のユ
 ーザによる第1のサブセットへの書込みアクセスのため
 の第1の要求を許可する手段と、他のユーザが第1のサ
 ブセットを変更することを禁止するために、データ・セ
 ットの第1のサブセットに書込みロックを設定し、他の
 ユーザによる第1のサブセットへの読出しアクセスは許
 可する手段と、を含むプロダクト。

(16) データ・セットの第2のサブセットへの書込み
 ロックを探索する手段と、第2のユーザによる第2のサ
 30 ブセットへの書込みアクセスのための第2の要求を許可
 する手段と、他のユーザが第2のサブセットを変更す
 ることを禁止するために、データ・セットの第2のサブ
 セットに書込みロックを設定し、他のユーザによる第2
 のサブセットへの読出しアクセスは許可する手段と、を
 含む、(15)記載のプロダクト。

(17) 第1のサブセット内の要素の第1のセット、及
 び第1のサブセット外の要素の第2のセットを有する、
 データ・セットの第3のサブセットへの書込みロックを
 探索する手段と、第3のユーザによる第3のサブセ
 40 ットの要素の第2のセットへの書込みアクセスのため
 の第3の要求を許可する手段と、第3のサブセットの要
 素の第2のセットへの書込みロックを設定する手段と、を
 含む、(15)記載のプロダクト。

(18) 第1のユーザからのコマンドに答して、第1
 のサブセットを変更する手段と、第1のサブセット内の
 変更を反映するために、データ・セットを更新する手段
 と、第1のサブセットへの書込みロックを解除する手段
 と、を含む、(15)記載のプロダクト。

(19) 第1のサブセットへの書込みロックが通知モー
 50

ドであり、第2のユーザによる第1のサブセットへの書
 込みアクセスのための第2の要求に答して、第1のサ
 ブセットへの書込みロックを探索する手段と、第1のユ
 ーザに、第1のサブセットに対する第2の要求が受信さ
 れたことを示すメッセージを送信する手段と、所定期間
 内に第1のユーザによる応答が受信されない場合、第1
 のサブセットへの書込みロックを解除する手段と、を含
 む、(15)記載のプロダクト。

(20) 第2のユーザによる第1のサブセットへの書込
 みアクセスのための第2の要求に答して、第1のサブ
 セットへの書込みロックを探索する手段と、第1のサブ
 セットへの書込みロックを見出したことに答して、
 予約キュー内に要求オブジェクトを配置する手段と、を
 含む、(15)記載のプロダクト。

(21) 第1のサブセットへの書込みロックが、第1の
 サブセットを記述するロック・キュー内のロック・オブ
 ジェクトである、(15)記載のプロダクト。

【0063】

【発明の効果】プログラマがオブジェクトの1部をチェ
 ック・アウトすることを許可する利点は、より多くのユ
 ーザが同一のオブジェクト上で作業することを可能とす
 る。しかしながら、同一オブジェクトに対する複数のユ
 ーザからの作業を一緒に併合することは、一般に困難な
 作業と認識されている。最適な同時制御はユーザがオブ
 ジェクトを自由にチェック・アウトすることを可能とす
 る。しかしながら、同一オブジェクトを更新する複数の
 ユーザが存在する場合、1人のユーザの更新だけが受諾
 される。他の全ての更新は打ち切られる。

【0063】以上述べたように、本願はバージョン化オ
 ブジェクトに対する改良されたロッキング機構を提案す
 る。

【図面の簡単な説明】

【図1】システム・ユニット、表示装置、キーボード、
 及びマウスを含む単一のコンピュータ・システムを表す
 図である。

【図2】図1に表されるコンピュータの構成要素のブロ
 ック図である。

【図3】図1のコンピュータのメモリ内のバージョン制
 御システムの構成要素のブロック図である。

【図4】バージョン制御システムの協調アクセス処理動
 作のブロック図である。

【図5】バージョン制御システムの協調アクセス処理動
 作のブロック図である。

【図6】バージョン制御システムの協調アクセス処理動
 作のブロック図である。

【図7】2人のユーザがあるオブジェクトの部分コピー
 をチェック・アウトし、第3のユーザが別の部分コピー
 のチェック・アウトを望む場合を示す図である。

【図8】あるユーザがオブジェクトの自身の部分コピー
 をチェック・インした前後における、図7のオブジェク
 50

21

トのロック・キューを示す図である。

【図9】通知ロック・モードの流れ図である。

【図10】予約ロック・モードの流れ図である。

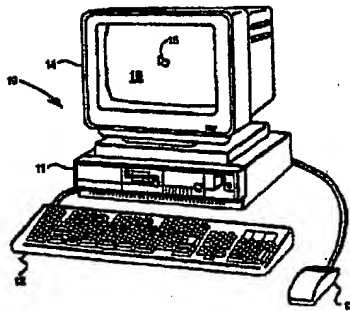
【図11】予約ロックモードにおける3つの要求オブジェクトを表す図である。

【符号の説明】

- 28 キーボード制御装置
29 マウス制御装置
30 ビデオ制御装置
40 入出力制御装置

10

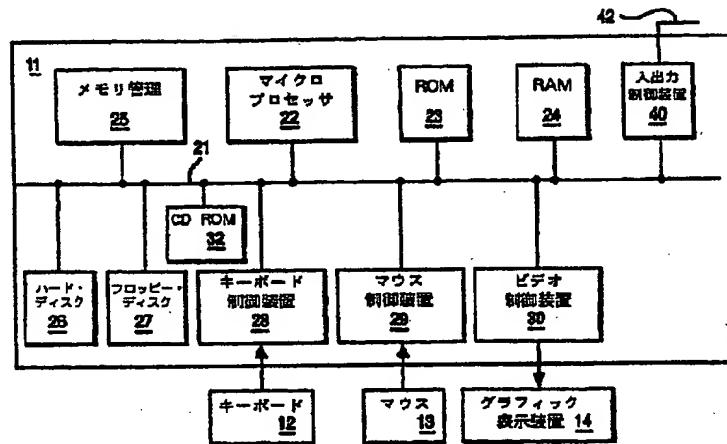
【図1】



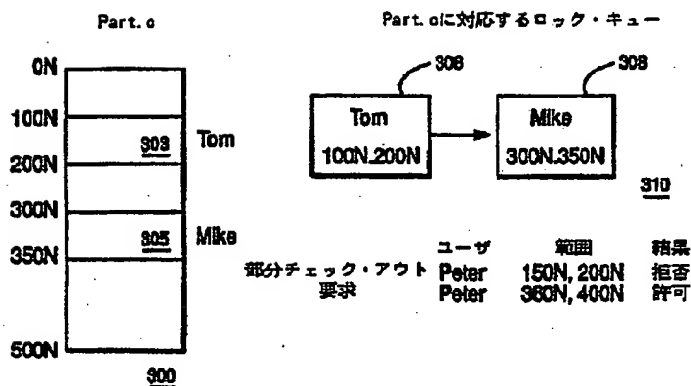
22

- 100 バージョン・マネージャ
101 ロック・マネージャ
102 固有ファイル・システム
103 チェック・アウト・キュー
104 予約済みのチェック・アウト・キュー
303、305 部分オブジェクト
306、308 ロック・オブジェクト
310 ロック・キュー
400 予約キュー

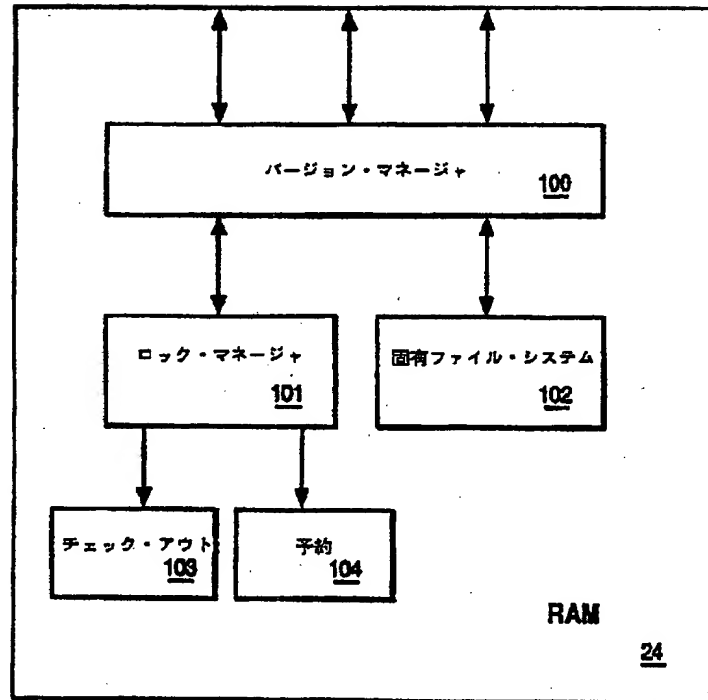
【図2】



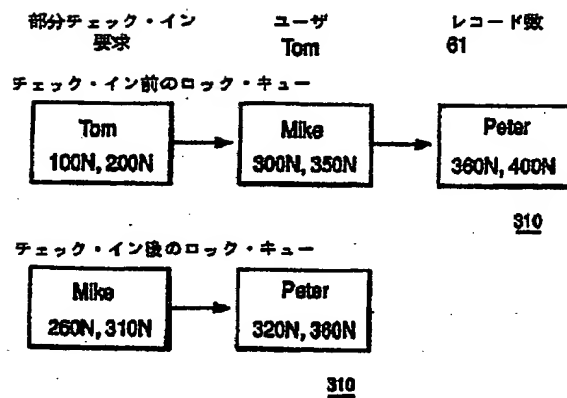
【図7】



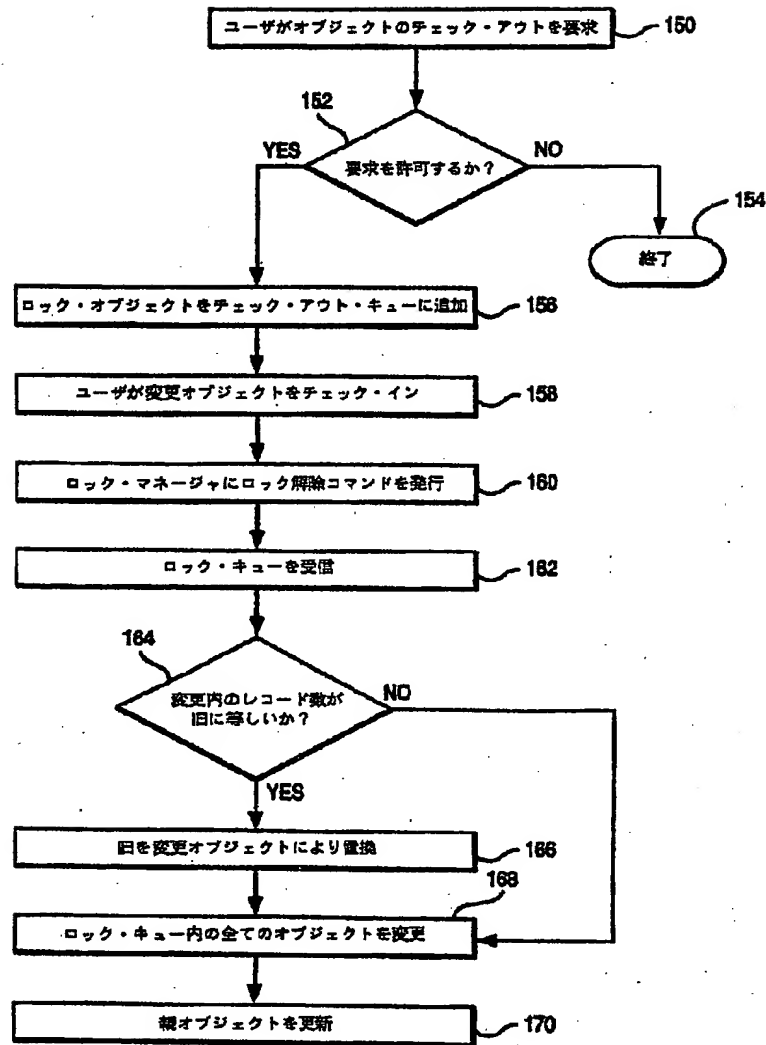
【図3】



【図8】

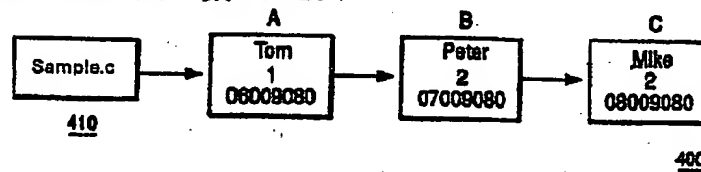


【図4】

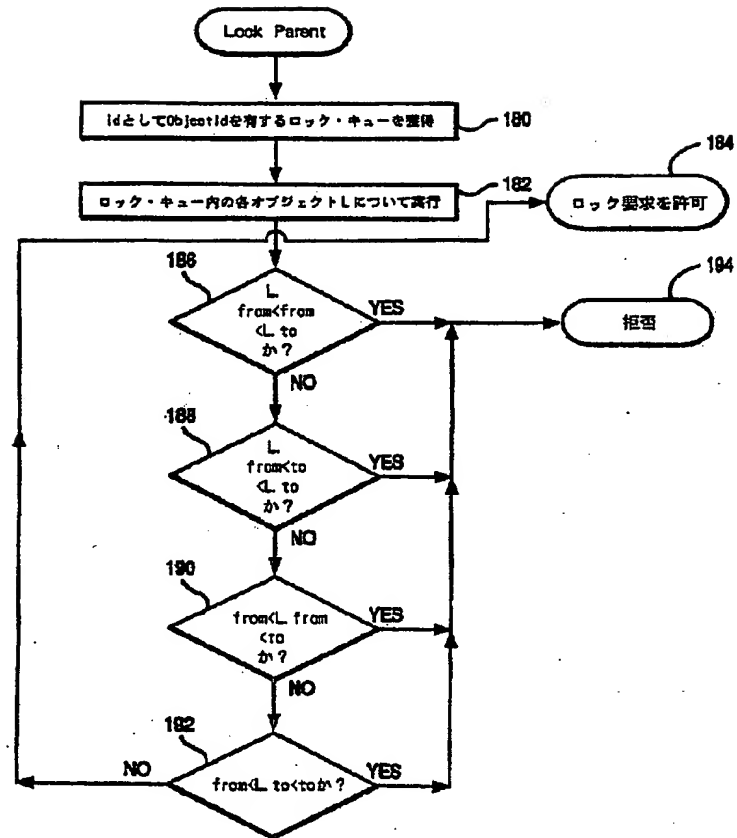


【図11】

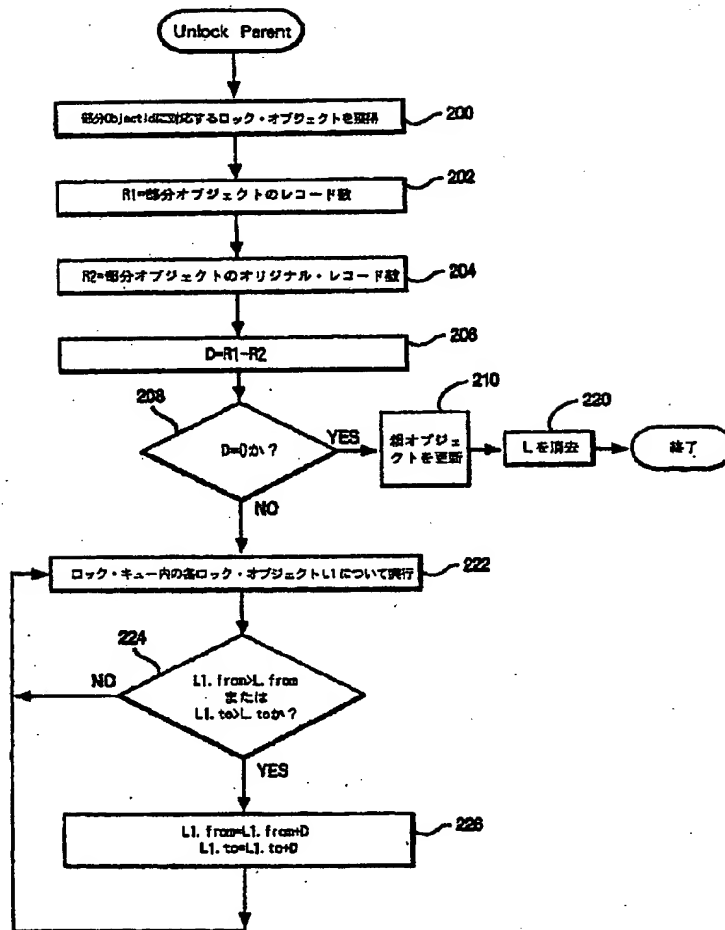
ソース・オブジェクト 要求オブジェクト



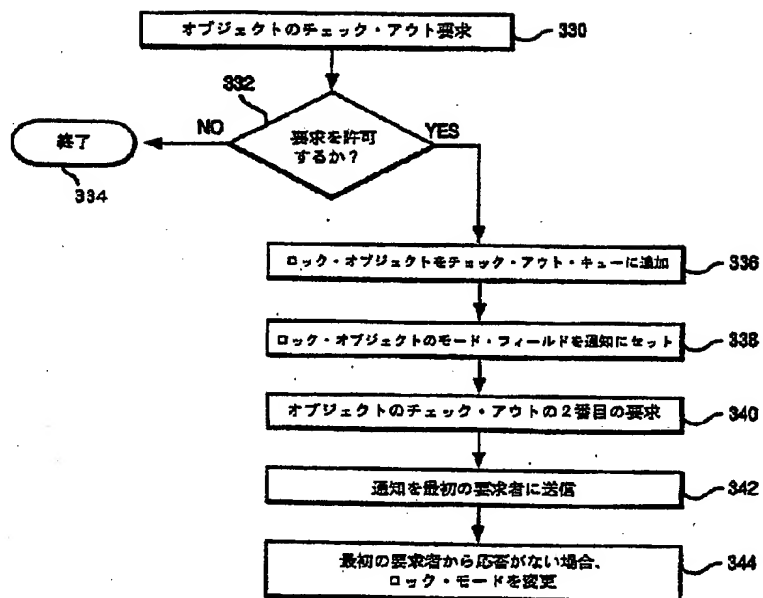
【図5】



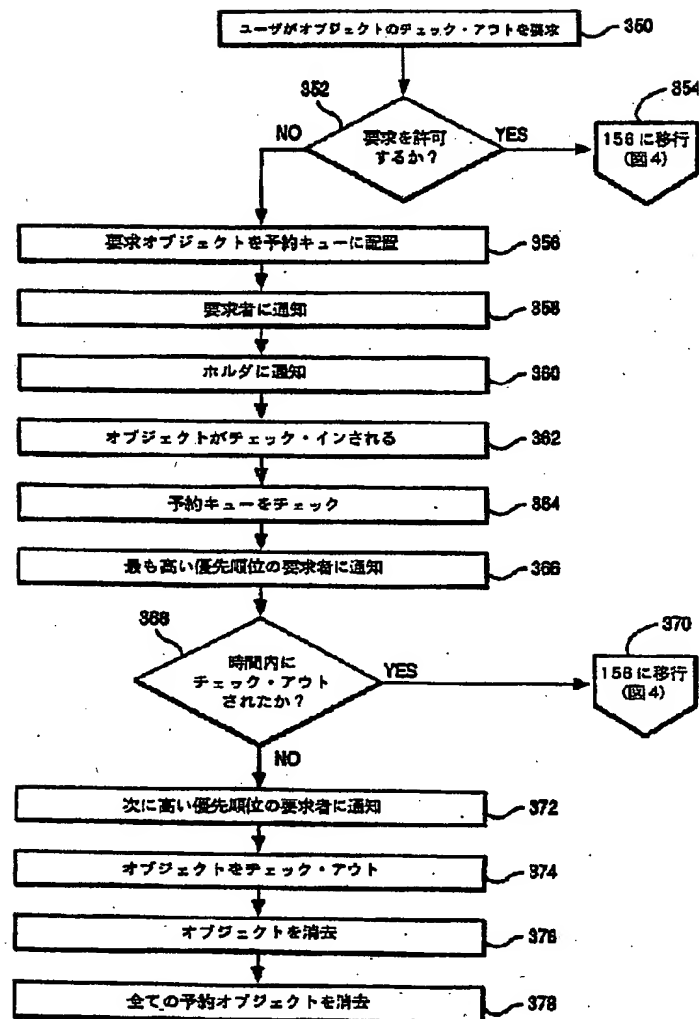
【図6】



【図9】



【図10】



フロントページの続き

(72)発明者 ピーター・チュン・ツイ・ウオン
 アメリカ合衆国78746、テキサス州オース
 テイン、キャノン・レン・ドライブ 6405